Exercise 5: Data Acquisition and File Exception Handling

Name - Anish Rao

Student Number - 20066423

Group - C

Colab Notebook Link

Python File and Exception Handling

1. Write a Python program to open a file named "example.txt" in write mode, write "Hello, World!" to it, and then close the file.

```
f = open("example.txt", "w")
f.write("Hello, World!")
f.close()
```

2. Modify the program in question 1 to handle the exception if the file cannot be opened.

```
try:
    f = open("example.txt", "w")
    f.write("Hello, World!")
    f.close()
except IOError:
    print("Error: Could not open or write to the file.")
```

3. Write a Python program to read the contents of "example.txt" and print them to the console. Use exception handling to catch any errors.

```
try:
    f = open("example.txt", "r")
    print(f.read())
except Exception as e:
    print("Error:", e)
finally:
    try:
        f.close()
    except:
        pass
```

```
Hello, World!
```

4. Write a Python program that attempts to open a non-existent file "missing.txt" and handles the FileNotFoundError gracefully.

```
try:
    f = open("missing.txt", "r")
    print(f.read())
except FileNotFoundError:
    print("File 'missing.txt' not found.")
```

```
File 'missing.txt' not found.
```

5. Create a Python program that writes a list of numbers (1 to 10) to a file, then reads the file and prints each number. Handle any potential exceptions.

```
try:
    with open("numbers.txt", "w") as f:
        for i in range(1, 11):
            f.write(str(i) + "\n")
    with open("numbers.txt", "r") as f:
```

```
        for line in f:
            print(line.strip())
except Exception as e:
    print("Error:", e)
```

```
1
2
3
4
5
6
7
8
9
10
```

6. Write a Python program that opens a file, reads its content line by line using a loop, and prints each line. Include exception handling.

```
try:
    f = open("example.txt", "r")
    for line in f:
        print(line.strip())
except Exception as e:
    print("Error:", e)
finally:
    try:
        f.close()
    except:
        pass
```

```
Hello, World!
```

7. Write a Python program that attempts to divide two numbers, taking user input, and handles the ZeroDivisionError if the user enters zero as the denominator.

```
try:
    num = float(input("Enter numerator: "))
    den = float(input("Enter denominator: "))
    print("Result:", num / den)
except ZeroDivisionError:
    print("Error: Cannot divide by zero.")
```

```
Enter numerator: 5
Enter denominator: 0
Error: Cannot divide by zero.
```

8. Modify question 7 to handle ValueError if the user inputs a non-numeric value instead of a number.

```
try:
    num = float(input("Enter numerator: "))
    den = float(input("Enter denominator: "))
    print("Result:", num / den)
except ZeroDivisionError:
    print("Error: Cannot divide by zero.")
except ValueError:
    print("Error: Please enter valid numeric values.")
```

```
Enter numerator: 5
Enter denominator: apple
Error: Please enter valid numeric values.
```

9. Write a Python program that attempts to open a file and uses a try-except-finally block to ensure the file is always closed properly.

```
try:
    f = open("example.txt", "r")
    print(f.read())
except Exception as e:
    print("Error:", e)
```

```
finally:
    try:
        f.close()
    except:
        pass
```

```
Hello, World!
```

10. Write a Python program that checks if a file exists before opening it. If the file exists, print its contents; otherwise, print a message stating that the file does not exist.

```
import os
filename = "example1.txt"
if os.path.exists(filename):
    with open(filename, "r") as f:
        print("Contents of", filename, ":\n", f.read())
else:
    print("File", filename, "does not exist.")
```

```
File example1.txt does not exist.
```

Data Acquisition

11. Write a Python program to read data from a CSV file named "data.csv" and print its contents.

```
import pandas as pd

try:
    df = pd.read_csv("data.csv")
    print(df.tail())
except Exception as e:
    print("Error reading data.csv:", e)
```

```
     Code Symbol                   Name
158  XOF    CFA  West African CFA franc
159  XPF     Fr               CFP franc
160  YER      ﷼             Yemeni rial
161  ZAR      R      South African rand
162  ZMW     ZK          Zambian kwacha
```

12. Write a Python program that reads an Excel file (data.xlsx) and prints the first five rows using the pandas library.

```
try:
    df = pd.read_excel("data.xlsx")
    print("First 5 rows:\n", df.head(5))
except Exception as e:
    print("Error reading data.xlsx:", e)
```

```
First 5 rows:
    Postcode  Sales_Rep_ID Sales_Rep_Name  Year         Value
0      2121           456           Jane  2011  84219.497311
1      2092           789         Ashish  2012  28322.192268
2      2128           456           Jane  2013  81878.997241
3      2073           123           John  2011  44491.142121
4      2134           789         Ashish  2012  71837.720959
```

13. Write a Python script to collect stock market data from an API (e.g., Alpha Vantage or Yahoo Finance) and display the latest stock price for a given company.

```
import yfinance as yf
ticker = input("Enter the stock ticker (e.g., AAPL): ")
try:
    stock = yf.Ticker(ticker)
    data = stock.history(period="1d")
    latest_price = data['Close'].iloc[-1]
    print("Latest price for", ticker, ":", latest_price)
except Exception as e:
    print("Error fetching stock data:", e)
```

```
Enter the stock ticker (e.g., AAPL): AAPL
Latest price for AAPL : 247.97000122070312
```

14. Write a Python program to check if a given dataset file (dataset.csv) exists before attempting to read it. If it doesn't exist, print a message informing the user.

```
filename = "dataset.csv"
if os.path.exists(filename):
    with open(filename, "r") as f:
        print("Contents of dataset.csv:\n", f.read())
else:
    print("File 'dataset.csv' does not exist.")
```

```
File 'dataset.csv' does not exist.
```

Web Scraping Questions

15. Write a Python script using requests and BeautifulSoup to extract the title of a webpage (e.g., "https://dbs.ie/").

Note - `https://dbs.ie/` was not working properly so used `https://met.ie/`

```
import requests
from bs4 import BeautifulSoup
url = "https://met.ie/"
try:
    response = requests.get(url)
    soup = BeautifulSoup(response.content, "html.parser")
    print("Page Title:", soup.title.string)
except Exception as e:
    print("Error scraping title:", e)
```

```
Page Title:
 Met Éireann – The Irish Meteorological Service
```

16. Write a Python program that scrapes all the links (< a > tags) from the webpage in question 16 above and prints them.

```
try:
    response = requests.get(url)
    soup = BeautifulSoup(response.content, "html.parser")
    links = soup.find_all("a")
    for link in links:
        print("Link:", link.get("href"))
except Exception as e:
    print("Error scraping links:", e)
```

```
Link: /
Link: #
Link: https://www.met.ie/warnings
Link: https://www.met.ie/forecasts/farming/graphs/fire-weather-index
Link: https://www.met.ie/forecasts/meteoalarm
Link: https://www.met.ie/forecasts/national-forecast
Link: https://www.met.ie/forecasts/listen-to-the-weather-forecast
Link: https://www.met.ie/forecasts/meteorologists-commentary
Link: https://www.met.ie/forecasts/national-forecast/
Link: https://www.met.ie/forecasts/atlantic-charts
Link: https://www.met.ie/forecasts/ensemble-maps
Link: https://www.met.ie/forecasts/marine-inland-lakes/sea-area-forecast-terminology
Link: https://www.met.ie/forecasts/marine-inland-lakes/sea-area-forecast
Link: https://www.met.ie/uv-index
Link: https://www.met.ie/forecasts/farming
Link: https://www.met.ie/forecasts/farming/agricultural-data-report
Link: https://www.met.ie/forecasts/storm-names
Link: https://www.met.ie/forecasts/mountains-forecast
Link: https://www.met.ie/forecasts/farming/graphs/fire-weather-index
Link: https://www.met.ie/forecasts/blight-forecast
Link: https://www.met.ie/forecasts/meteoalarm
Link: https://www.met.ie/forecasts/worldweather
Link: #
Link: https://www.met.ie/latest-reports/observations
Link: https://www.met.ie/latest-reports/observations/yesterday
```

```
Link: https://www.met.ie/latest-reports/surface-analysis
Link: https://www.met.ie/latest-reports/satellites
Link: https://www.met.ie/latest-reports/satellites/ireland-infrared
Link: https://www.met.ie/latest-reports/satellites/europe-infrared
Link: https://www.met.ie/latest-reports/satellites/world-infrared
Link: https://www.met.ie/wow
Link: https://www.met.ie/latest-reports/recent-rainfall-radar
Link: https://www.met.ie/latest-reports/recent-rainfall-radar/12-hour-rainfall-radar
Link: https://www.met.ie/latest-reports/recent-rainfall-radar/24-hour-rainfall-radar
Link: https://www.met.ie/latest-reports/recent-rainfall-radar/48-hour-rainfall-radar
Link: https://www.met.ie/forecasts/marine-inland-lakes/buoys
Link: https://www.met.ie/latest-reports/valentia-tephigram
Link: #
Link: https://www.met.ie/climate/climate-of-ireland
Link: https://www.met.ie/climate/climate-change
Link: https://www.met.ie/climate/weather-extreme-records
Link: https://www.met.ie/climate/major-weather-events
Link: https://www.met.ie/climate/summer-centre
Link: https://www.met.ie/climate/storm-centre
Link: https://www.met.ie/climate/past-weather-statements
Link: https://www.met.ie/climate/past-weather-statements/past-weather-agrometeorological-bulletins
Link: https://www.met.ie/climate/services
Link: https://www.met.ie/nfcs
Link: https://www.met.ie/wow
Link: https://www.met.ie/climate/available-data
Link: https://www.met.ie/climate/available-data/daily-data
Link: https://www.met.ie/climate/available-data/historical-data
Link: https://www.met.ie/climate/available-data/monthly-data
Link: https://www.met.ie/climate/available-data/mera
Link: https://www.met.ie/climate/available-data/long-term-data-sets
Link: https://www.met.ie/climate/30-year-averages
Link: https://www.met.ie/climate/climate-change-indices-etccdi
Link: https://www.met.ie/climate/available-data/climate-data-for-thermal-modelling-of-buildings
```

17. Write a Python script that extracts and prints all headings (h1, h2, and h3 tags) from the given webpage in question 16 above.

```
try:
    response = requests.get(url)
    soup = BeautifulSoup(response.content, "html.parser")
    for tag in ["h1", "h2", "h3"]:
        headings = soup.find_all(tag)
        for heading in headings:
            print(f"{tag}:", heading.get_text(strip=True))
except Exception as e:
    print("Error scraping headings:", e)
```

```
h2: National Forecast
h2: Met News
h2: Search
h2: Connect
h2: Download our apps
h3: Climate Statement for January 2025
h3: Ballinrobe Students Scoop Met Éireann Award at BTYSTE
h3: Second AMOC Research Meeting to take place in Dublin
h3: Annual Climate Statement for 2024
h3: Agriculture
h3: Marine
h3: Monthly Reports
h3: Storm Names
h3: Be Winter Ready
```

18. Write a Python script that scrapes a table from the webpage in question 16 above and stores the data in a CSV file.

```
try:
    response = requests.get(url)
    soup = BeautifulSoup(response.content, "html.parser")
    table = soup.find("table")
    if table:
        rows = table.find_all("tr")
        with open("table_data.csv", "w", newline="") as csvfile:
            writer = csv.writer(csvfile)
            for row in rows:
                cols = row.find_all(["th", "td"])
                cols_text = [col.get_text(strip=True) for col in cols]
                writer.writerow(cols_text)
        print("Table data has been written to table_data.csv")
    else:
```

```
        print("No table found on the webpage.")
except Exception as e:
    print("Error scraping table:", e)
```

```
No table found on the webpage.
```